**МИНОБРНАУКИ РОССИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №2  
дисциплина: Вычислительная математика  
тема: «Интерполяция функций»

Выполнил: ст. группы ВТ-22  
Воскобойников Илья Сергеевич  
Проверил: Бондаренко Т.В.

Белгород 2019

**Цель работы:** изучить способы построения интерполяционного многочлена для случая равномерной и неравномерной сетки интерполяции, получить практические навыки решения задачи интерполяции.

Вариант 4



Задания к работе:

1. Найти область допустимых значений переменной х для функции y= f(x) задания соответствующего варианта.

2. Составить таблицу значений функции y = f(x), используя (n ≥ 6) узлов интерполяции (x i ≠ a, где a точка, не являющаяся узлом интерполяционной сетке, в которой необходимо приближенно вычислить значение функции в соответствии с вариантом задания; x 0 <a< x n ).

3. По полученной таблице значений функции y = f(x) составить интерполяционный многочлен Лагранжа для случаев линейной, квадратичной и кубической интерполяции: L 1 (x), L 2 (x), L 3 (x).

Замечание. Интервал (x 0 , x n ), n = 1, 2, 3, используемый для построения интерполяционного многочлена Лагранжа должен содержать точку a.

4. По таблице значений функции составить интерполяционный многочлен Ньютона In (x). При построении интерполяционного многочлена Ньютона необходимо использовать конечные разности для случая равномерной сетки интерполяции и разделенные разности для неравномерной сетки интерполяции. Можно построить таблицу значений функции для равномерной сетки, выполнить построение многочлена Ньютона с конечными разностями, затем убрать 1 значение из середины таблицы и выполнить построение многочлена Ньютона с разделенными разностями для получившейся неравномерной сетки интерполяции.

5. Вычислить точное значение функции y = f(x) при x=a (y T = f(a)).

6. Вычислить приближенное значение функции при x=a по всем полученным интерполяционным многочленам.

7. Определить абсолютную ∆ и относительную δ погрешность вычисления значения функции для каждого интерполяционного многочлена (интерполяционные многочлены Лагранжа и Ньютона) при заданном значении х=а.

8. Построить в одной системе координат графики полученных интерполяционных функций (многочлены Лагранжа и Ньютона), исходной функции y= f(x) и отметить значения функций в точке x=a.

9. Представить полученные результаты в виде таблицы (см. табл. 1)

10. Составить программу, реализующую вычисление приближенного значения функции в произвольной точке путем построения интерполяционного многочлена Ньютона для случая равномерной и неравномерной сетки.

Задание 1

D(y) : (-∞;+∞)

Задание 2

Таблица значений

|  |  |  |
| --- | --- | --- |
| i | xi | yi |
| 0 | 1 | 3.0174 |
| 1 | 12 | 3.2078 |
| 2 | 23 | 3,3905 |
| 3 | 34 | 3.5589 |
| 4 | 45 | 3.7068 |
| 5 | 56 | 3.8287 |
| 6 | 67 | 3.9202 |
| 7 | 78 | 3.9659 |

3+sin(46)?

Задание 3,5,6,7

**Линейная интерполяция**

45<46<56

X0=45; y0=3.7068

X1=56; y1=3.8287

L1(x) = y0l0(1)(x)+ y1l1(1)(x) =y0 + y1

L1(x) = 3.7178

∆=|yт-упр|

δ=|\*100%

3+sin(46) = 3.90179

∆=|L1(46)- 3+sin(46)|=0.18399

δ=|\*100% = 0.0471

**Квадратичная интерполяция**

Х0= 45 ; x1=56 ; x2=67

y0= 3.0174; y1=3.2078; y2=3,3905

L2(x)=y0l0(2)(x)+y1l1(2)(x)+y2l2(2)(x)

L2(x)= y0

L2(x)=3.93503

∆= 0.186676

δ=0.02215

**Кубическая интерполяция**

Х0= 45 ; x1=56 ; x2=67x3=78

y0= 3.0174; y1=3.2078; y2=3,3905; y3=3.9659

L3(x)=y0l0(3)(x)+y1l1(3)(x)+y2l2(3)(x) y3l3(3)(x)

L3(x)= y0

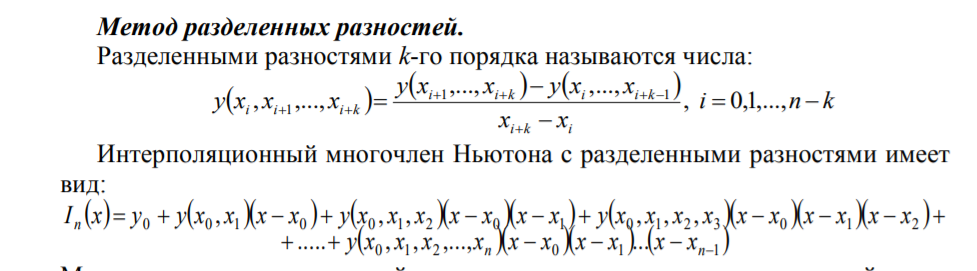
L3(x)=3.9555

∆= 0.18762

δ=0.0219

**Задание 4**

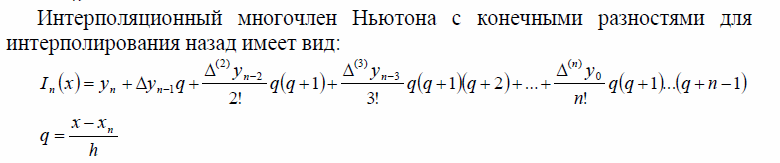
**Раздельные разности:**



∆=|L1(46)-3+sin(46)|= 0.058

δ= 0.015 %

**Конечные разности:**



= - 4.09

∆=|L1(46)-3+sin(51)|= 0,04140

δ = 0.0125

Таблица 1

Оценка погрешности интерполяционного многочлена

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Многочлен Лагранжа | | | Многочлен Ньютона | |
| Погрешность | линейная  интерполяция | квадратичная  интерполяция | кубическая  интерполяция | разделенные  разности | конечные  разности |
| ∆ | 0.18399 | 0.186676 | 0.8762 | 0.058 | 0,04140 |
| δ | 0.0471 | 0.02215 | 0.0219 | 0.015 | 0.0125 |

**Задание 8**

#include **<stdio.h>**#include **<stdlib.h>  
  
void** sort(**float** \*\*a, size\_t n)  
{  
 **float** \*x = malloc((n+1)\***sizeof**(**float**\*));  
 **for** (**int** i = 0; i <= n; i++)  
 **for** (**int** j = n; j > i; j--)  
 **if** (a[j-1][0] > a[j][0])  
 {  
 x = a[j-1];  
 a[j-1] = a[j];  
 a[j] = x;  
 }  
}  
  
**int** check(**float** \*\*a, size\_t n)  
{  
 **int** i = 2,f = 1;  
 **float** x = a[1][0], h = a[1][0] - a[0][0];  
 **while** (i <= n && f)  
 {  
 f = a[i][0] - x == h;  
 x = a[i][0];  
 i++;  
 }  
 **return** f;  
}  
  
**float** raz\_razn(**float** \*\*a, size\_t n, **float** x)  
{  
 **for** (**int** i = 2; i < n + 2; i++)  
 **for** (**int** j = 0; j < n - i + 2; j++)  
 a[j][i] = (a[j+1][i-1] - a[j][i-1])/(a[j+i-1][0] - a[j][0]);  
 **double** s = 0, p = 1, f = 1;  
 **for** (**int** i = 1; i < n + 2; i++)  
 {  
 p = 1;  
 **for** (**int** k = 0; k < i - 1; k++)  
 p = p\*(x-a[k][0]);  
 s = s + a[0][i]\*p;  
 }  
 **return** s;  
}  
  
**int** fact(**int** n)  
{  
 **int** p = 1;  
 **for** (**int** i = 1; i <= n; i++)  
 p = p \* i;  
 **return** p;  
}  
  
**double** kon\_razn(**float** \*\*a, size\_t n, **float** x)  
{  
 **for** (**int** i = 2; i < n + 2; i++)  
 **for** (**int** j = 0; j < n - i + 2; j++)  
 a[j][i] = (a[j+1][i-1] - a[j][i-1]);  
 **float** q = (x - a[0][0])/(a[1][0] - a[0][0]);  
 **double** s = 0, p = 1;  
 **for** (**int** i = 1; i < n + 2; i++)  
 {  
 p = 1;  
 **for** (**int** k = 0; k < i - 1; k++)  
 p = p\*(q-k);  
 s = s + a[0][i]/fact(i-1)\*p;  
 }  
 **return** s;  
}  
  
**void** write\_matr(**float** \*\*a, size\_t n)  
{  
 **for** (**int** i = 0; i <= n + 3 - i; i++)  
 {  
 **for** (**int** j = 0; j < n-i+2; j++)  
 printf(**"%.2f\t"**,a[i][j]);  
 printf(**"\n"**);  
 }  
}  
  
**int** main()  
{  
 size\_t n;  
 scanf(**"%i"**,&n);  
 **float** \*\*a = malloc((n+1)\***sizeof**(**float**\*));  
 **float** x;  
 **double** s;  
 **for** (**int** i = 0; i <= n; i++)  
 {  
 a[i] = malloc((n+3)\***sizeof**(**float**));  
 scanf(**"%f %f"**,&a[i][0],&a[i][1]);  
 }  
 scanf(**"%f"**,&x);  
 sort(a,n);  
 **if** (check(a,n))  
 s = kon\_razn(a,n,x);  
 **else** s = raz\_razn(a,n,x);  
 write\_matr(a,n);  
 printf(**"I = %lf"**,s);  
 **for** (**int** i = 0; i <= n; i++)  
 free(a[i]);  
 free(a);  
}